Parsing Pose of People with Interaction

Serim Ryou
sryou@caltech.edu
Pietro Perona
perona@caltech.edu

Computational Vision Lab
California Institute of Technology
Pasadena, CA, USA

Abstract

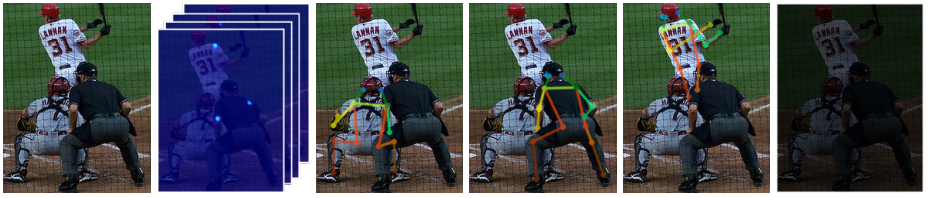
We propose an end-to-end multi-person pose estimation model that learns to predict keypoint locations for each person in the scene, regardless of the complexity of their social interactions. While recent multi-person pose estimation algorithms achieve high performance on scenes where people do not overlap, these algorithms produce undesired outcomes, *e.g.* merging two people or swapping similar parts of different people, when the people in the scene are heavily occluded. To attack this issue, we have curated a subset of COCO [20] containing such scenes and call it COCO-crowd. We formulate multi-person pose estimation as a sequential prediction problem that first generates heatmaps of the potential part locations and then assembles the parts into separate instances, each representing a single person, using convolutional LSTMs. Despite using a small-scale dataset (relative to all of COCO), we achieved comparable performance to state-of-the-art methods trained on the full COCO dataset. We also evaluate our method on the Immediacy dataset [7], which consists of images with diverse social interactions, *e.g.* standing shoulder to shoulder, or hugging, and achieve state-of-the-art results.

1 Introduction

Pose estimation, localizing joint locations in an input image, is an important building block for high level computer vision tasks such as human action recognition [34, 36], human re-identification [40] and proxemics inference [7, 39]. Multi-person pose estimation focuses on predicting a distinct keypoint skeleton for each person in an input image. Recent multi-person pose estimation methods produce promising results with deep convolutional neural networks and large-scale datasets [20]. These methods are categorized into 1) *top-down approaches* [6, 10, 15, 25] that independently run single-person pose estimation algorithms subsequent to human detection results, and 2) *bottom-up approaches* [3, 17, 23, 27] that group the estimated joint locations into instances representing individual people.

Although current pose algorithms work well for scenes with minimal occlusion, it is still a challenging problem to cluster the correct parts in cluttered scenes. In scenes with human interaction, multi-person pose estimation becomes a challenging task, as body parts are often partially occluded and/or intertwined. These scenarios have been identified as a challenge by the pose-estimation community, and methods have been suggested to improve performance. For instance, the winning entry of 2017 COCO keypoint challenge [6] (a top-down approach) defines "hard" keypoints and performs a refinement process on the initial prediction in difficult cases. Bottom-up approaches have suggested methods that attempt

© 2018. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.



(a) Input (b) Part heatmaps (c) Corresponding ordered outputs of the model

Figure 1: **An Overview of our system.** For an input image (a), part localizer (Section 4.1) produces keypoint heatmaps without identity (b). Person decoding module (Section 4.2) sequentially produces instance keypoint heatmaps with distinct person identities. The network ends the prediction with generating the all-zero heatmap which represents no more instances on the scene (c).

to learn additional cues for the succeeding inference step, *i.e.*, pairwise terms [17], identity embedding [23], or part affinity field [3]. However, these methods ignore the possibility that multiple people can have the same part located at the same image coordinate, which is more likely in highly crowded scenes.

Ronchi and Perona [30] provided in-depth analysis of the performance drops on pose estimation algorithms; they concluded that state of the art methods vastly underperform in crowd scenes, because parts are mostly occluded due to overlapped instances. In addition, since crowd scenarios are rarely found in COCO, which is the most commonly used pose estimation dataset, the community has suffered from scarcity of appropriate training data for developing accurate multi-person pose estimation algorithms.

In this paper, we aim to develop a multi-person pose estimation algorithm that is able to decouple human poses despite considerable overlaps between interacting instances. COCO dataset has non-overlapping instance bias, which we discuss in section 3. To overcome this bias, we analyze the COCO keypoint dataset and extract all images which show overlapped instances. Also, we revisit the Immediacy dataset [2] which contains images with significant overlap between people. We propose a single-pipeline framework that is trainable in a fully end-to-end fashion. Unlike [23], our method directly renders the final instance heatmaps so that an additional inference step is unnecessary. We let the network have the global encoding of the entire scene and sequentially recognize individual instances to improve the overall pose estimation performance in crowd scenarios. Storing the memory of the entire scene and the histories of the instances, we empower the network to handle occluded parts in the overlapped areas. Figure 1 illustrates an overview of our system.

We summarize the main contributions of this paper as follows:

- We tackle a challenging problem in multi-person pose estimation that deals with the severe overlaps arisen from human interactions; and

- We achieve comparable performance with the state-of-the-art methods despite training with significantly less data.

(a) COCO (b) COCO-Crowd

Figure 2: **Crowd Extraction Procedure.** The left image (a) shows an original COCO image and annotations, and the two images on the right (b) show the corresponding images in COCO-crowd. We define the notion of "overlap" when the intersection of union (IoU) score of two bounding boxes is greater than 0.1 ([1,2], [3,4], and [4,5] pairs on left image). All interlinked boxes ([1,2] and [3,4,5]) are merged into proposals for the crowd region. Meanwhile, we discard non-crowd regions, box without an overlap.

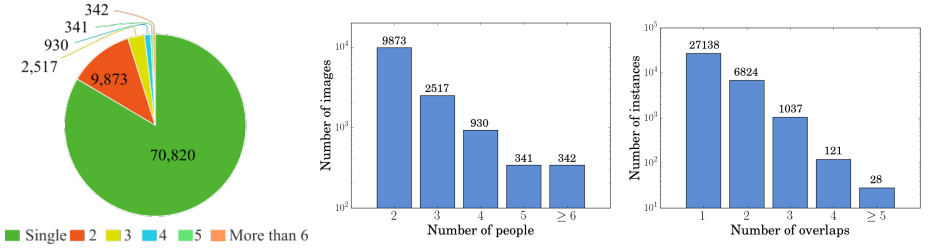
2 Related work

Single-Person Pose Estimation. Earlier approaches in pose estimation [5, 18, 26, 38] employ graphical models, where each node represents a keypoint and each edge encodes limb information. Deformable Part Models (DPM) [11] decompose objects into parts and use spatial relations among the parts to build computationally tractable inference steps. With the advent of deep convolutional neural networks (DCNN), researchers began to apply DCNNs for keypoint feature extraction and limb representation. Chen *et al*. [5] define limb configurations using pairwise clusters of adjacent keypoints, and employ a DCNN to extract the unary and pairwise scores for each keypoint. A single unified model [33] was proposed to combine a DCNN part detector with a spatial model enforcing implicit constraints on the body parts.

Bulat *et al*. [2] proposed a cascaded CNN architecture that performs regression on the first predicted part heatmap. Several approaches [4, 22, 35] exploit iterative refinements and show significant improvement. In particular, the stacked hourglass network [22] consists of repeating multi-scale modules and performs sequential refinements to capture complex spatial relationships. Chu *et al*. [8] exploited attention mechanisms at multiple resolutions and applied Conditional Random Fields (CRF) to model the correlations in neighboring regions. Yang *et al*. [37] proposed a pyramid residual module on skip connections of the hourglass block to learn multi-scale features.

Multi-person Pose Estimation. Current multi-person pose estimation methods can be classified into two main categories: *top-down approaches* [6, 10, 15, 25] and *bottom-up approaches* [3, 17, 23]. Top-down approaches first detect candidate human bounding boxes, then run a single-person pose estimation algorithm on each box. Papandreou *et al*. [25] followed this two-step pipeline with Faster-RCNN [28] as a human detector and fully convolutional ResNet [14] as a pose estimator. Fang *et al*. [10] proposed a symmetric spatial transformer network to produce a high quality single-person region. Mask-RCNN [15] proposed a framework for both instance segmentation and pose estimation by predicting an object mask and keypoint locations in parallel with the existing branch for bounding box recognition.

On the other hand, bottom-up approaches first predict part locations, then assemble the



(a) (b) (c)

Figure 3: **Dataset configuration.** After running our crowd extraction system on COCO, we observe that single-instances are dominant on the dataset (a). To overcome this dataset bias, our dataset (named COCO-crowd) consists of images with two or more instances, where its distribution can be seen in (b). To show the complexity of the our curated dataset, we count the number of overlaps arisen from each instance, and provide the distribution in (c).

parts into distinct people. Pishchulin *et al*. [27] proposed a partitioning and labeling formulation based on the CNN part detectors and Integer Linear Programming (ILP). DeeperCut [17] extended this work by incorporating image-conditioned pairwise probabilities that consider body part configurations into the deep network. Cao *et al*. [3] exploited a two-stage pipeline, which first generated part heatmaps and part affinity fields along the limbs, and then assigned part identity through a bipartite graph matching algorithm. Newell *et al*. [23] proposed an end-to-end system which directly output part identity tags along with the part locations.

Recurrent Model with Spatial Sequence Prediction. Our work formulates multi-person pose estimation as a sequential problem using spatial variants of recurrent neural networks. Gkioxari *et al*. [13] adopted a sequential model for single-person pose estimation by predicting each joint location dependent on the previous output, allowing the network to learn complex body structure. Shi *et al*. [31] proposed the Convolutional LSTM (ConvLSTM), a convolutional variant of the standard LSTM [16], to capture spatiotemporal correlation within precipitation forecasting. Romera-Paredes and Torr [29] proposed a class-specific instance segmentation and counting method by sequentially segmenting one instance of the scene at a time using ConvLSTM.

3 COCO-Crowd dataset

Our work focuses on parsing the poses of people in crowd scenes. With this perspective, we analyze the COCO dataset. Previous work on COCO keypoint evaluation [30] defines "overlap" between instances if a pairwise instance shows an intersection over union (IoU) score greater than 0.1. We borrow this definition to extract regions of images that exhibit overlap to use in our dataset. In order to discover these regions, we iterate over all possible pairs of bounding boxes containing a person in each image. If a pair of boxes have IoU$\geq$0.1, then we tag that pair of boxes as a crowd. After all crowd pairs are obtained, we merge all pairs that share at least one common instance into sets. Figure 2 describes this process in detail. We also summarize the resulting data distribution in Figure 3. Our dataset, named as COCO-crowd, has 14,003 training images containing 35,148 total instances. Test and validation images are also produced by following this procedure on 2014 COCO validation data, which results in 3,336 validation and 3,336 test images.

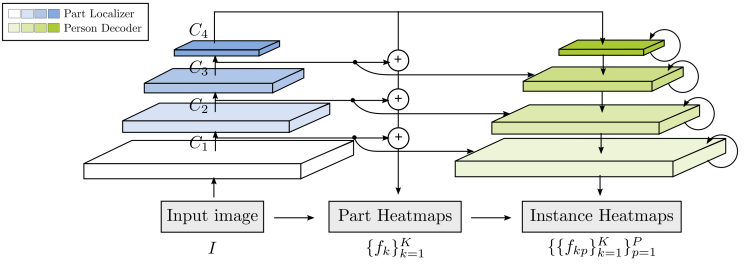


Figure 4: **Network architecture.** The network consists of two parts: part localizer (each blue box representing ResNet-50 convolution block) and person decoder (green boxes corresponding to ConvLSTM block at each resolution). The input image is encoded with part localizer and first predicts part heatmaps. The person decoder decouples this encoded feature into distinct instance heatmaps.

4 Method

Figure 4 provides an overview of our system composed of two parts: part localizer and person decoder. The proposed framework is a single pipeline that encodes the input image to predict K keypoint heatmaps using a fully convolutional network, and sequentially produces instance heatmaps using a convolutional recurrent neural network. We use ConvLSTM in order to decode the individual instances. We describe the details of the part prediction in Section 4.1 and explain how we apply ConvLSTM to our framework in Section 4.2.

4.1 Part localizer

We use ResNet-50 [14] as our building block for keypoint detection. Similar to [6, 21], we utilize feature pyramid structure to preserve both semantic information and the localization quality. An input image I is encoded with ResNet conv blocks, and transformed into feature maps in different scales as C_1, C_2, C_3, and C_4, respectively (see Figure 4). We apply 1×1 kernel convolution to match the dimension of the all feature maps to 64. Then, we resize and sum these feature maps to produce the final part heatmap. We apply a sigmoid to the summed feature maps. The output of the part localization module has the form of K heatmaps, each representing a single part location, with an output stride of 4. We denote the final output heatmap as $f_k(x_i)$, where k represents the k-th keypoint (out of K) and $x_i \in \{1,\ldots,N\}$ represents the index of 2D pixel location.

4.2 Person decoder

We model multi-person pose estimation as a sequential prediction problem with variable length of output. In our problem setting, the model should keep track of the number of people, and individuate an instance from a set of human candidates. Since pose estimation requires high localization quality, we adopt a spatial variant on LSTM, ConvLSTM. To preserve multi-resolution information, we apply ConvLSTM units at every scale encoded from the part localization step.

The architecture of the person decoding module is displayed in Figure 5. The person decoder consists of a chain of ConvLSTMs at every scale. All of the ConvLSTM kernels are 3×3. The features C_1, C_2, C_3, and C_4 generated from the part localization module are

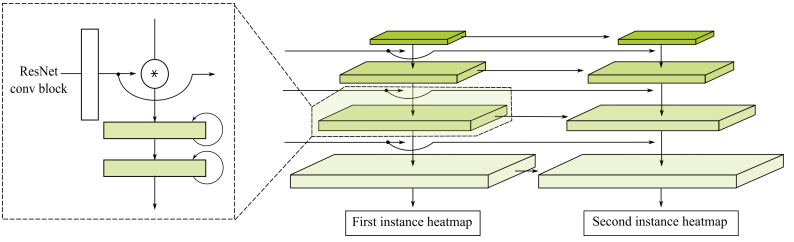


Figure 5: **Person Decoding Module.** The green block represents two ConvLSTM layers at each resolution. The recurrent block on each scale is composed of two stacked ConvLSTMs.

fed through all the subsequent recurrent stages to prevent the network from forgetting keypoint information. In particular, we halve the dimensionality of part features by applying one convolutional block of ResNet, which we explain the details in the supplementary material. These features are concatenated to the input for each ConvLSTM block. We employ two stacked ConvLSTM layers for each scale block, so that the output from the first ConvLSTM acts as an input to the second unit. After passing the ConvLSTM block, features are upsampled by 2, and the final output of person decoder has an output stride of 4, producing K keypoint heatmaps of the target person. We denote the output as $f_{kp}(x_i)$ where $p \in \{1,\ldots,P\}$ represents p-th person over the total number of people P in an input image. In the same manner as the part localization step, we apply a sigmoid on top of the outputs. When the network finishes prediction, it is trained to output an all-zero heatmap. We provide the details of the person decoder in the supplementary material.

4.3 Loss function

In our experimental setting, the network first predicts candidate part locations and finally produces a heatmap for each instance. Thus, the loss function consists of two parts.

Part localization. Let x_i be a 2D location on the image, where $i \in \{1,\ldots,N\}$ indexing the pixel locations. For each part type $k \in \{1,\ldots,K\}$, we denote $h_k(x_i)$ as the k-th keypoint heatmap at location x_i. The ground truth heatmap $h_k(x_i) = 1$, when $\|x_i - y\| \leq R$ for $y \in \{y_{k0}, y_{k1}, \ldots, y_{kP}\}$, each y_{kp} representing part-k location of the p-th person over all P people, and zero otherwise. In our experiments, we set $R = 3$ pixels. This part heatmap encodes all part locations without identity. We apply pixelwise binary cross entropy to the output of the part localization module with $h_k(x_i)$. The part localization loss is as follows:

$$\mathcal{L}_{part} = \frac{1}{NK} \sum_{k=1}^{K} \sum_{i=1}^{N} \mathcal{L}_{BCE}(f_k(x_i), h_k(x_i)), \tag{1}$$

where $\mathcal{L}_{BCE}$ denotes pixelwise binary cross entropy.

Person decoder. Let $h_{kp}(x_i)$ be the k-th keypoint heatmap of p-th person at location x_i. The keypoint heatmap $h_{kp}(x_i) = 1$ when $\|x_i - y_{kp}\| \leq R$ with y_{kp} ground truth location of part-k of p-th person, and zero otherwise. The network produces set of keypoint heatmaps at each step, encoding part locations of each person. In order to make the network decide the order in which to predict each instance, we use the Hungarian algorithm [19], as in [29, 32]. Given a cost matrix, the Hungarian algorithm finds an optimal matching between the output and the target heatmaps and re-orders the target heatmaps in a matched order. We construct our cost matrix by computing binary cross entropy for each prediction-target pair. Given the

Method	AP	AP.5	AP.75	AP M	AP L	AR	AR.5	AR.75	AR M	AR L
Mask-RCNN [13]	0.364	0.598	0.362	0.371	0.398	0.497	0.706	0.519	0.505	0.528
CMU-pose [3]	0.365	0.599	0.369	0.367	0.378	0.418	0.628	0.429	0.422	0.438
AE [23]	**0.438**	0.664	**0.456**	**0.440**	0.451	**0.532**	0.740	**0.560**	**0.538**	**0.554**
AE*	0.396	0.663	0.402	0.409	0.420	0.486	0.728	0.507	0.493	0.515
Ours	0.433	**0.709**	0.447	**0.440**	**0.454**	0.520	**0.761**	0.549	0.526	0.545

Table 1: **Results (AP) on COCO-crowd.** Mask-RCNN is tested using Detectron [12] and all other methods are tested using the code and pretrained models the authors provide. Testing is held on single scale on all bottom-up methods. To see the impact of the amount of data, we also trained associative-embedding [23] on COCO-crowd (AE*).

re-ordered heatmaps from the Hungarian algorithm, we again apply binary cross entropy in order to compute our loss. We additionally apply loss for the following two steps, as in [29], with zero heatmaps, so that the network learns the stop criterion.

$$\mathcal{L}_{person} = \frac{1}{NK(P+2)} \sum_{p=1}^{P+2} \sum_{k=1}^{K} \sum_{i=1}^{N} \mathcal{L}_{BCE}(H(f_{kp}(x_i), h_{kp}(x_i))), \tag{2}$$

where $H(\cdot)$ denotes the Hungarian algorithm, which returns the re-ordered target and input. The final loss is as follows:

$$\mathcal{L} = \lambda_0 \, \mathcal{L}_{part} + \mathcal{L}_{person}, \tag{3}$$

where $\lambda_0 = 0.5$ is a hyperparameter which controls the relative importance of two terms.

5 Experimental results

5.1 Experimental setup

Training Setup. We have implemented our system in PyTorch. We optimize eq. (3) with Adam and train for 130 epochs. For COCO-crowd, the learning rate is set to 1e-3 and is decayed by 0.1 at epoch 60 and 90, respectively. With the same initial setting, the learning rate is dropped by 0.1 at 40 and 60 for the Immediacy dataset. We use a batch size of 32 on 8 GPUs for COCO-crowd, whereas a batch size of 12 on a single GPU for Immediacy. For the part encoding backbone (*i.e.*, ResNet), we employ the initial weights pretrained on ImageNet [9]. The input size is set to 512×512. We augment the data with random flips, rotations ($\pm 40^\circ$), and scalings on the fly. When training the model on COCO-crowd, we use the corresponding original COCO images for the scale augmentation to contain various backgrounds. The cropped box is enlarged when the scaling factor is greater than 1.0.

We follow the curriculum learning scheme used in [1, 29] by gradually increasing number of people after the loss converges. Therefore, the network learns to predict at most M instances in iteration M, even when more instances are present. In our experiments, we train the network to predict at most two people until convergence, then increase the maximum number of people by 1 every two epochs. For COCO-crowd, the loss is masked to avoid penalizing instances without annotation.

Testing Setup. Testing is performed on a single scale with both the original and a flipped version of each image. If the maximum value of the heatmap is less than the threshold (0.05), the network produces all-zero heatmap to stop the prediction.

Method	2	3	4	5	≥ 6
AE [23]	0.503	0.435	0.367	**0.405**	**0.419**
Ours	**0.512**	**0.439**	**0.393**	0.386	0.364

Table 2: **AP score by number of people**

Jittering (px)	± 0	± 5	± 10	± 15
AP	0.433	0.426	0.412	0.392

Table 3: **AP score by jittering bounding box**

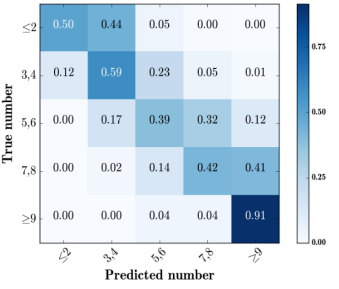


Figure 6: **Counting confusion matrix**

5.2 Evaluation

5.2.1 COCO-crowd

COCO keypoint dataset has 17 keypoint labels of 5 facial landmarks (nose, left/right ear and eye) and 12 body parts (left/right shoulder, elbow, wrist, hip, knee and ankle). We report the performance with three different algorithms using the official evaluation metric, average precision (AP) and average recall (AR) in Table 1. Two bottom-up methods [3, 23] are tested in a single scale using the code and pretrained models that the authors provide. Mask-RCNN [15] is tested using Detectron [12] with the ResNeXt-101 encoding backbone, which showed the highest mAP score among all detectron models.

With a small amount of data, we outperform two different methods and achieve comparable performance to the state-of-the-art method trained on the full COCO dataset. To see the impact of the amount of data, we also train AE [23] from scratch, using COCO-crowd (AE*). When compared against state-of-the-art methods trained on the same amount of data, our method shows promising results. We also perform an additional experiment to gauge the importance of part localization module. When training without the part localization loss, we observed a huge performance drop, AP score of *0.271* compared to the original score *0.433*.

Counting. To see if our method successfully learns when to stop, we visualize the confusion matrix for the number of predictions and the number of ground truth instances in Figure 6. We observe that most of the elements are in diagonal, which implies our method can approximately count the number of people in an image.

Performances over the number of people. We evaluate the score by varying number of people in an image and compare against the state-of-the-art method in Table 2. Our method performs better on predicting relatively small number of people. Due to the recurrent architecture, we observed failure cases as the sequence length increases.

Bounding box jittering test. COCO-crowd dataset is composed of the cropped regions for the crowd, thus it requires crowd detections in advance. To show the feasibility of our framework as a full system, we show how our method is robust at bounding box jittering in Table 3. While some part locations can be eliminated from the bounding box as jittering, our method faithfully estimates pose of people in the box.

5.2.2 Immediacy Dataset

The Immediacy dataset was originally designed to analyze visual interaction between people. In this dataset people are mostly present in pairs, either holding one another from behind, hugging, holding hands, giving each other a high five or putting arms over each other's

Figure 7: **Qualitative results.** Results containing severe occlusion due to social interaction.

shoulders. It contains 7,500 training images and 2,500 testing images. We used 500 images from the training set for validation. The total number of instance is 20,499, each having 12 keypoint labels of upper body (head top, neck, left/right shoulder, left/right elbow, left/right wrist, left/right hand, left/right hip). This dataset is challenging for inferring the arm locations, since social interaction makes significant arm occlusions.

We followed the percentage of correct keypoints (PCK) metric [38] used in the dataset paper [7]. PCK measure is for a single-person pose estimation problem, where an estimated body part location is defined to be correct when it falls within α $max(height, width)$ pixels. We used $\alpha = 0.2$ as in the original setup of [7]. Since the results from the paper are evaluated given the bounding box of ground truth upper body, we match the result to corresponding ground truth and report the mean PCK of matched keypoints, for fair comparison. To show how current methods perform at this dataset, we also test Mask-RCNN [15] and report score of the parts in common. Even without using the person location, current methods significantly outperform all previous methods. In particular, our method improves wrist and hand predictions by a wide margin. Even without exhibiting the Immediacy dataset, our model trained on COCO-crowd still shows huge performance gains on wrist compared to Mask-RCNN. We provide qualitative results on both datasets in Figure 7.

Method	head	shoulder	elbow	wrist	hand	torso	mean
Yang *et al.* [33]	69.5	63.0	42.6	31.8	29.0	43.9	47.0
Ouyang *et al.* [24]	67.7	61.3	46.4	35.4	32.5	48.9	49.0
Chu *et al.* [4]	82.5	74.6	50.1	38.8	37.1	55.4	56.4
Mask-RCNN [15]	-	81.0	64.3	55.3	-	-	-
Ours (crowd)	-	86.8	65.5	63.7	-	-	-
Ours	**95.6**	**88.8**	**72.4**	**74.0**	**73.3**	**75.1**	**79.9**

Table 4: **PCK score on Immediacy Dataset**.

6 Conclusions

In this paper, we addressed the problem of pose estimation in crowd scenes and proposed a multi-person pose estimation method which sequentially decouples each instance. We tested our approach with two challenging datasets and showed that the proposed method is able to infer human poses regardless of complex interactions. With considerably small amount of data, our method achieved a comparable performance to the state-of-the-art method trained on the full COCO dataset. Furthermore, we significantly improved the performance on the Immediacy dataset, containing heavily occluded scenes due to social interactions, and produced faithful predictions on the arm locations. We believe our approach to be applicable to general multi-person pose estimation followed by crowd detection.

Acknowledgement

This work was supported by funding from Disney, Inc and Allen E. Puckett Endowment Fund.

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009.

[2] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, 2016.

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *Proc. IEEE CVPR*, 2017.

[4] João Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proc. IEEE CVPR*, 2016.

[5] Xianjie Chen and Alan Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014.

[6] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. *CoRR*, abs/1711.07319, 2017.

[7] Xiao Chu, Wanli Ouyang, Wei Yang, and Xiaogang Wang. Multi-task recurrent neural network for immediacy prediction. In *Proc. IEEE ICCV*, 2015.

[8] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proc. IEEE CVPR*, 2017.

[9] Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE CVPR*, 2009.

[10] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *Proc. IEEE ICCV*, 2017.

[11] P. Felzenszwalb, D. Mcallester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. IEEE CVPR*, 2008.

[12] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018.

[13] Georgia Gkioxari, Alexander Toshev, and Navdeep Jaitly. Chained predictions using convolutional neural networks. In *ECCV*, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, 2016.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *Proc. IEEE ICCV*, 2017.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[17] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016.

[18] Martin Kiefel and Peter Gehler. Human pose estimation with fields of parts. In *ECCV*, 2014.

[19] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97, 1955.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[21] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *Proc. IEEE CVPR*, 2017.

[22] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[23] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NIPS*, 2017.

[24] Wanli Ouyang, Xiao Chu, and Xiaogang Wang. Multi-source deep learning for human pose estimation. In *Proc. IEEE CVPR*, 2014.

[25] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Christoph Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proc. IEEE CVPR*, 2017.

[26] Leonid Pishchulin, Micha Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Proc. IEEE CVPR*, 2013.

[27] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V. Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proc. IEEE CVPR*, 2016.

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*. 2015.

[29] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *ECCV*, 2016.

[30] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *Proc. IEEE ICCV*, Oct 2017.

[31] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

[32] Russell Stewart, Mykhaylo Andriluka, and Andrew Y. Ng. End-to-end people detection in crowded scenes. In *Proc. IEEE CVPR*, 2016.

[33] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014.

[34] Chunyu Wang, Yizhou Wang, and Alan L. Yuille. An approach to pose-based action recognition. In *Proc. IEEE CVPR*, 2013.

[35] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proc. IEEE CVPR*, 2016.

[36] Lu Xia, Chia-Chih Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPR Workshops*, 2012.

[37] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *Proc. IEEE ICCV*, 2017.

[38] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE TPAMI*, 35, 2013.

[39] Yi Yang, Simon Baker, Anitha Kannan, and Deva Ramanan. Recognizing proxemics in personal photos. In *Proc. IEEE CVPR*, 2012.

[40] Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *Proc. IEEE CVPR*, 2017.